



## UNF Digital Commons

---

UNF Graduate Theses and Dissertations

Student Scholarship

---

2014

# Enabling One-Phase Commit (1PC) Protocol for Web Service Atomic Transaction (WS-AT)

Chirag N. Rana  
*University of North Florida*

---

### Suggested Citation

Rana, Chirag N., "Enabling One-Phase Commit (1PC) Protocol for Web Service Atomic Transaction (WS-AT)" (2014). *UNF Graduate Theses and Dissertations*. 498.  
<https://digitalcommons.unf.edu/etd/498>

This Master's Thesis is brought to you for free and open access by the Student Scholarship at UNF Digital Commons. It has been accepted for inclusion in UNF Graduate Theses and Dissertations by an authorized administrator of UNF Digital Commons. For more information, please contact [Digital Projects](#).

© 2014 All Rights Reserved



ENABLING ONE-PHASE COMMIT (1PC) PROTOCOL FOR  
WEB SERVICE ATOMIC TRANSACTION (WS-AT)

by

Chirag N. Rana

A thesis submitted to the  
School of Computing  
in partial fulfillment of the requirements for the degree of

Master of Science in Computing and Information Sciences

UNIVERSITY OF NORTH FLORIDA  
SCHOOL OF COMPUTING

March, 2014

Copyright (©) 2014 by Chirag N. Rana

All rights reserved. Reproduction in whole or in part in any form requires the prior written permission of Chirag N. Rana or designated representative.

The thesis “Enabling One-Phase Commit (1PC) Protocol for Web Service Atomic Transaction (WS-AT)” submitted by Chirag N. Rana in partial fulfillment of the requirements for the degree of Master of Science in Computing and Information Sciences has been

Approved by the thesis committee:

Date

---

Dr. Karthikeyan Umapathy  
Thesis Advisor and Committee Chairperson

---

Dr. Asai Asaithambi

---

Dr. Lakshmi Goel

Accepted for the School of Computing:

---

Dr. Asai Asaithambi  
Director of the School

Accepted for the College of Computing, Engineering, and Construction:

---

Dr. Mark Tumeo  
Dean of the College

Accepted for the University:

---

Dr. Len Roberson  
Dean of the Graduate School

## ACKNOWLEDGEMENT

I would like to express my special appreciation to my advisor, Dr. Karthikeyan Umapathy for being a tremendous mentor for me. I would like to thank him for all his advice and inputs regarding my thesis. His advice on this thesis and the suggestions to improve materials presented were of great help. I would like to thank Dr. Asai Asaithambi and Dr. Lakshmi Goel, for serving as my committee members even through their busy schedules, and for the suggestions and advice that were tremendously helpful to complete this thesis.

I am extremely grateful to the JBoss Team - Dr. Paul Robinson (Web Service Transaction Lead at JBoss) and Mr. Mark Little (Architect at JBoss) for their advice and inputs on conceptual and technical aspects of this thesis. Their inputs were of great help for the technical implementation of my thesis.

A special thanks to Mr. Jim Littleton for reviewing my thesis report and ensuring it is in compliance with thesis document requirements. I am also indebted to Dr. Roger Eggen for navigating me through the Graduate School and School of Computing processes. In addition, I want to give special thanks to the School of Computing at University of North Florida for allowing me to use all the required resources to complete this thesis.

Last but not the least, I would like to express appreciation and thanks to my beloved wife, Jigisha, for her personal support and great patience at all times.

## CONTENTS

FIGURES .....	vii
TABLES .....	viii
EQUATIONS .....	ix
ABSTRACT .....	x
Chapter 1. Introduction .....	1
1.1    Problem Statement .....	4
Chapter 2. Background and Literature review .....	7
2.1    Transactions .....	7
2.1.1    Two-Phase Commit (2PC) Protocol .....	8
2.1.2    One-Phase Commit (1PC) Protocol .....	9
2.1.3    Compare and Contrast 2PC and 1PC .....	10
2.2    Web Service .....	10
2.3    WS-Atomic Transactions (WS-AT) .....	11
2.4    WS - Coordination (WS-C) .....	13
2.5    WS-Business Activity .....	14
2.6    JBoss Transactions .....	15
2.7    Related Work .....	16
Chapter 3. 1PC for WS-AT .....	21
3.1    Architecture Diagram of the Proposed Solution .....	22
3.2    Selection of Implementation Platform .....	24

3.3	Implementation of 1PC Mechanism .....	25
Chapter 4. Research Methodology .....		31
4.1	Design Science Research Methodology .....	31
4.2	Design Science Research Guidelines .....	32
Chapter 5. Evaluation .....		35
5.1	Experimental Setup Scenarios .....	37
5.1.1	Scenario 1: Balance Inquiry in Online Banking .....	37
5.1.2	Scenario 2: Bill Pay .....	38
5.1.3	Scenario 3: Restaurant Table Booking.....	39
Chapter 6. Evaluation Results .....		41
6.1	Independent Samples t-Test.....	41
6.2	Effect Size .....	52
6.3	Power Analysis.....	55
Chapter 7. Concluding Remarks .....		57
References .....		59
Vita.....		63

## FIGURES

Figure 1.	Two-Phase Commit (2PC) Protocol.....	9
Figure 2.	2PC state transition diagram in WS-AT.....	11
Figure 3.	WS-Coordination Specification.....	14
Figure 4.	Architecture Diagram of the Proposed Mechanism.....	23
Figure 5.	Sequence Diagram for existing 2PC mechanism.....	29
Figure 6.	Sequence Diagram for new 1PC mechanism.....	30
Figure 7.	BPMN Diagram for Inquiring Bank Account Balance process .....	38
Figure 8.	BPMN Diagram for Bill Paying process.....	39
Figure 9.	BPMN Diagram for Restaurant Table Booking process .....	40



## TABLES

Table 1.	Group Statistics for Scenario 1 - Online Balance Inquiry .....	44
Table 2.	Independent Samples t-Test Results for Scenario 1 with 50 Users.....	44
Table 3.	Independent Samples t-Test Results for Scenario 1 with 100 Users.....	45
Table 4.	Independent Samples t-Test Results for Scenario 1 with 150 Users.....	45
Table 5.	Independent Samples t-Test Results for Scenario 1 with 200 Users.....	46
Table 6.	Group Statistics for Scenario 2 - Online Bill Pay .....	47
Table 7.	Independent Samples t-Test Results for Scenario 2 with 50 Users.....	47
Table 8.	Independent Samples t-Test Results for Scenario 2 with 100 Users.....	48
Table 9.	Independent Samples t-Test Results for Scenario 2 with 150 Users.....	48
Table 10.	Independent Samples t-Test Results for Scenario 2 with 200 Users.....	49
Table 11.	Group Statistics for Scenario 3 - Online Restaurant Table Booking .....	50
Table 12.	Independent Samples t-Test Results for Scenario 3 with 50 Users.....	50
Table 13.	Independent Samples t-Test Results for Scenario 3 with 100 Users.....	51
Table 14.	Independent Samples t-Test Results for Scenario 3 with 150 Users.....	51
Table 15.	Independent Samples t-Test Results for Scenario 3 with 200 Users.....	52
Table 16.	Effect Size and Post Hoc Power for all three Scenarios .....	54

## EQUATIONS

Equation 1.	Cohen's Effect Size .....	53
-------------	---------------------------	----

## ABSTRACT

Business transactions (a.k.a., business conversations) are series of message exchanges that occur between software applications coordinating to achieve a business objective. Web service has been proven to be a promising technology in supporting business transactions. Business transaction can either be long-running or short-lived. A transaction whether in a database or web service paradigm consists of an “all-or-nothing” property. A transaction could either succeed or fail. Web Service Atomic Transactions (WS-AT) is a specification that currently supports Two-Phase Commit (2PC) protocol in a short-lived transaction. WS-AT is developed by OASIS—a standards development organization. However, not all business process scenarios require a 2PC, in that case, just a One-Phase Commit (1PC) would be sufficient. But unfortunately, WS-AT currently does not support 1PC optimization.

The ideal scenario where 1PC can be used instead of 2PC is when there is only a single participant. Short-lived transactions involving only one participant can commit without requiring initial “prepare” phase. Thus, there is no overhead to check whether the participant is prepared to either commit or rollback. This research focuses on designing a mechanism that can add 1PC support in WS-AT. The technical implementation of this mechanism is developed by using JBoss Transaction API. As a part of this thesis, 1PC mechanism for a single participant scenario was implemented. This mechanism optimizes the web service transaction process in terms of overhead and performance in terms of

execution time. The technical implementation solution for 1PC mechanism was evaluated using three different business process scenarios in a controlled experiment as a presence or absence test. Evaluation results show that 1PC mechanism has a lower mean for execution time and performed significantly better than 2PC mechanism. Based on the contributions made by this thesis, we recommend OASIS to consider including 1PC mechanism as a part of the WS-AT specification.

## Chapter 1

### INTRODUCTION

Business transactions (a.k.a., business conversations) are series of message exchanges that occur between software applications coordinating to achieve a business objective (Papazoglou, 2003). The conversations in Business-to-Business (B2B) paradigm are often complex involving many participants within a network or cross-networks. The majority of the B2B conversations requires transactional support, which guarantees the correct order of execution and desired results (Bowles & Moschoyiannis, 2008). The web service has been proven to be a promising technology in supporting business transactions (Bowles & Moschoyiannis, 2008). Service-oriented architecture (SOA) is an architectural approach for the implementation and delivery of loosely coupled distributed services.

A business transaction can either be long-running or short-lived. Transactions that can be executed within a few minutes, hours, or even a few days are known as long-running (Bowles & Moschoyiannis, 2008). Long-running transactions are required in cases where conversations are complex and consist of multiple business activities. Business activities typically consist of a series of smaller sub-transactions within one complex transaction or just consist of a single transaction. Such transactions originate from different sources which have multiple web services running to achieve a specific result. In contrast, short-running conversations are required to achieve a specific single unit of task or atomic task ((Little, Maron, & Pavlik, 2004), pp. 32). Short-running conversations are short-lived transactions

which are usually atomic in nature. A long-running conversation can be a series of multiple short-running conversations.

For example, consider a trip booking scenario. Let us say a person wants to book a flight and a rental car. Step one would be to reserve a flight followed by a rental car reservation as the second step. Let us say that step two fails, then it is not feasible to let go of the flight booking as it may be full when trying to reserve the next time. In that case, the user can reserve a rental car successfully through another agency. As a compensation step, if the user finds another cheap flight he may cancel the previously booked flight and book a new one. This is an example of a long-running transaction with compensation. It consists of two short-lived transactions.

Let us consider another example which explains short-lived transactions. Let us say a person wants to book three tickets for a music concert for his family with desired seat numbers after reviewing the diagram of available seats. In this case the system should allow him to book all the three tickets within one transaction which can be considered as an atomic transaction. It would either book all three tickets in the case of success of a transaction or none in the case of a transaction failure. Both of the above mentioned types of transactions are important in complex business scenarios.

Regardless of the transaction types and business scenarios, short execution time (a.k.a., response time) is of high importance for online business transactions as a few seconds could be intolerable for a human user (Shneiderman, 1984; Singhal, 1988; Srinivasan,

Anderson, & Ponnnavolu, 2002). A customer has varied options in online environment, thus, a frustrated customer would likely switch to a competitor's service (Srinivasan, et al., 2002). Customers expect online transaction processing to be fast and efficient (Constantinides, 2004). Therefore, transactions with shorter execution times have a higher likelihood of maintaining customer loyalty and satisfaction. Customer satisfaction and loyalty have been recognized as important factors that affect the profitability of a business (Srinivasan, et al., 2002).

There are different web service specifications which support long-running and short-lived transactions. Long-running transactions are supported by several competing specifications such as Web Service Choreography Description Language (WS-CDL) (Kavantzas et al., 2005), Web Service Business-Activity (WS-BA) (Newcomer, Robinson, Freund, & Little, 2007), Business Transaction Protocol (BTP) (Ceponkus et al., 2002), and Web Service Composite Application Framework (WS-CAF, 2005). Short-lived transactions are supported by WS-Atomic Transaction (WS-AT) specification (Newcomer, Robinson, Little, & Wilkinson, 2009). WS-AT specification is similar to traditional ACID transactions. All the above mentioned specifications are OASIS standard except WS-CDL which is a W3C standard. OASIS and W3C are standard development organizations that utilize consensus-oriented process to bring various industry members and experts together to design web standards including web service specifications (Umapathy, Purao, & Bagby, 2012).

This thesis mainly focuses on short-lived transactions and WS-AT specification because in a short-lived transaction, the resources are blocked until the transaction ends either successfully or in a failure state. As discussed earlier, that transaction is atomic in nature providing an “all-or-nothing” property. This property is very useful in many real world applications where ACID (atomicity, consistency, isolation and durability) properties are key requirements. WS-AT specification supports short-lived transactions. There are many research opportunities in this area which can prove useful in the real world while designing applications which are transactional in nature.

### 1.1 Problem Statement

Two-Phase Commit Protocol (2PC) is a widely accepted industrial standard to maintain the atomicity of a transaction. This protocol is an agreement amongst the members participating in a transaction. The 2PC means that the transaction manager first sends out a “prepare” message to all participants and starts waiting for acknowledgement messages (Newcomer, Robinson, Little, et al., 2009). Once it receives “OK” from every participant, it sends out a “commit” message. If it didn't receive an “OK” from some or all participants, it sends out a “rollback” message to all participants.

If a transaction involves more than one resource, 2PC is necessary. The 2PC protocol (the “prepare” phase and the “commit” phase) ensures that when the transaction ends, all changes to all resources are either totally committed or fully rolled back. All the participants are then informed of the final result. This is the reason why 2PC is needed in



distributed transactions. For example, in case of ordering a book scenario, you might have two separate participants (one adding a book to a shopping cart and another, a payment process) within a single transaction. If the first process has been committed but the second fails, there is no way to roll back the first one anymore.

While 2PC ensures atomicity (a single, indivisible atomic unit of work that either commits or rolls back), it comes with a high cost of performance hit due to the number of message exchanges and the logging of states which are used for further processing. For this reason several optimizations of the protocol or even completely new solutions are required. One-Phase Commit (1PC) is a widely known optimization (Neto & Reverbel, 2008). When talking about a single phase commit, the transaction manager only sends out one message, “commit”. It does not send “prepare for commit” message. This reduction in the overhead of sending “prepare for commit” message could potentially increase the performance of the transaction manager. It also reduces the chances of failure that could occur during the “prepare for commit” phase.

WS-AT (Web service Atomic Transactions) is a specification developed by OASIS that currently supports 2PC protocol in any short-lived transaction (Newcomer, Robinson, Little, et al., 2009). There are some scenarios such as business processes with a single participant, where it is not necessary to have a 2PC. In that case, there should be a mechanism to support 1PC. Unfortunately, WS-AT specification currently does not support 1PC optimization. WS-AT specification developed by OASIS follows “Design by Committee” process (Purao, Bagby, & Umapathy, 2008), thus, in the due process some

functionalities are considered and some are not included due to various technical and political reasons (Little, 2007).

The scenario where 1PC can suffice the desired result of a transaction is when it has only a single participant. In a current situation, even if a distributed transaction involves only a single participant, WS-AT requires execution of the full 2PC protocol. The lack of the 1PC in WS-AT is unfortunate, since it is an important and widely known optimization for performance and overhead of transaction processing.

The objective of this thesis is to enable 1PC in WS-AT so that in case of a single participant, there is no overhead to check whether the participant is prepared to commit or rollback. It can do it without requiring the initial “prepare” phase. 1PC is an optimization selected by the coordinator when it observes that only one participant has been registered for the transaction. In many cases, the participant is not aware of number of participants registered with the coordinator for a transaction. The participant should notify the coordinator that it is capable of participating in a 1PC or 2PC protocol. The coordinator would then select the 1PC optimization if there is only one participant registered for the transaction and only if that participant is 1PC capable. In order for coordinator to allow individual participants to register for different protocols, there is a need to add a 1PC mechanism for WS-AT. This thesis provides a conceptual model of the 1PC mechanism, prototype for the mechanism, and preliminary evaluation of the prototype.

## Chapter 2

### BACKGROUND AND LITERATURE REVIEW

#### 2.1 Transactions

A transaction, whether in a database or web service paradigm, consists of an “all-or-nothing” property ((Little, et al., 2004), pp.4). Transaction could either succeed or fail. In the case of success, it will give the desired result and it reaches a state which can be called as a success state. But in the case of failure, it will either revert to its original starting point or achieve a new state which can be stated as a failure state depending upon the design of an application.

As mentioned earlier, web service transactions are similar to database transactions possessing similar properties. In the real-world environment, transactions are needed to perform critical tasks like airline ticket reservation, online money transfers in a bank, etc. There is a possibility that an operation can go into an inconsistent state if it is not bound into a transaction. Let us take a look at important properties of transaction which are also known as ACID Properties ((Little, et al., 2004), pp.6):

- **Atomicity:** In case of a successful completion of transaction, it commits. If a transaction fails, it reaches its origin state which is also called rollback.
- **Consistency:** In case of success or failure of transaction, the data will remain consistent all the time.

- **Isolation:** Even if transactions are executed concurrently, the results obtained at the end of each transaction are such that it appears to have been executed serially.
- **Durability:** Once the transactions is completed successfully and committed, the effect is permanent.

Every transaction has a coordinator which manages the outcome of the transaction (success/commit or failure/rollback). The coordinator is also known as Transaction Manager.

#### 2.1.1 Two-Phase Commit (2PC) Protocol

Two-Phase Commit protocol is a widely accepted industrial standard to maintain the atomicity of a transaction. This protocol is an agreement amongst the members participating in a transaction. Figure 1 shows the phases of 2PC protocol (Dinn, Connor, & Little, 2014).

As shown in Figure 1, a transaction first enters into phase one. The coordinator C will start a conversation with the participants A and B enlisted in a transaction. Based on the response from the participants, it decides whether to commit or rollback. If both participants agree to commit, the coordinator will remember the decision and the transaction enters into phase two. During this phase, the coordinator will inform the participants to carry out the action whether to commit or rollback depending upon the response it gets during the first phase.

Image redacted, paper copy available upon request to home institution.

**Figure 1. Two-Phase Commit (2PC) Protocol**

2.1.2 One-Phase Commit (1PC) Protocol

One-Phase Commit protocol is an optimization of 2PC protocol. Business transaction involving a single participant can use 1PC instead of a standard 2PC. In this protocol, there is only one phase which is the second phase of 2PC protocol. Here is a simple example which explains 1PC. Consider a transaction in which a user wants to reserve a table at a restaurant. In this case, the user simply goes to the restaurant's website and books a table. In this scenario, only one participant is involved, so 1PC is a perfect fit.

### 2.1.3 Compare and Contrast 2PC and 1PC

2PC maintains atomicity in business transaction, while 1PC does the same thing but with optimization. 2PC is used in transaction in a multi-participant scenario while 1PC is used in transaction having a single participant. While 2PC ensures atomicity (a single, indivisible atomic unit of work that either commits or rollbacks), it comes with a high cost of performance hit due to the number of message exchanges and the logging of states which are used for further processing. When talking about a single phase commit, the transaction manager only sends out one message, “commit”. It does not send “prepare for commit” message. This reduces the overhead of exchange for “prepare for commit” message could increase the performance. It also reduces the chances of failure that could occur during “prepare for commit” phase.

## 2.2 Web Service

“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.” (Booth et al., 2004). Web service is one of the widely used technologies to achieve communication between two or more participants within the same network or across the networks (Booth, et al., 2004).

### 2.3 WS-Atomic Transactions (WS-AT)

WS-Atomic Transaction is similar to traditional transaction with ACID properties. This specification is an OASIS standard. This specification is generally used for short-running conversations. Figure 2 shows the 2PC state transitions (Newcomer, Robinson, Little, et al., 2009).

Image redacted, paper copy available upon request to home institution.

**Figure 2. 2PC state transition diagram in WS-AT**

The participant accepts:

**Prepare:**

When a coordinator sends this notification, the participant will enter the first phase and vote on the outcome of the transaction. A participant that is in its Active state should send a vote as “Aborted”, “Prepared”, or “ReadOnly”. If the participant is unaware of the

transaction, it must send an “Aborted” notification and if it has already voted then it must resend the same vote.

**Rollback:**

When a coordinator sends this notification, the participant is aware that it has to abort and forget the transaction. If a participant is not committing then it must respond by sending an “Aborted” notification and should then forget about the transaction. If the participant is unaware of the transaction, it must send an “Aborted” notification to the coordinator.

**Commit:**

When a coordinator sends this notification, the participant is aware that it has to commit the transaction. This notification must only be sent after the “prepare” phase and if the participant voted to commit. If the participant is unaware of the transaction, it must send a “Committed” notification to the coordinator.

The coordinator accepts:

**Prepared:**

When a participant sends this notification, the coordinator has the knowledge that the participant is Prepared and votes to commit the transaction.

**ReadOnly:**

When a participant sends this notification, the coordinator has the knowledge that the participant votes to commit the transaction, and has forgotten the transaction. The participant does not wish to participate in phase two.



**Aborted:**

When a participant sends this notification, the coordinator has the knowledge that the participant has aborted and forgotten the transaction.

**Committed:**

When a participant sends this notification, the coordinator has the knowledge that the participant has committed and forgotten the transaction.

## 2.4 WS - Coordination (WS-C)

WS-Coordination (WS-C) specification defines an extensible framework for coordinating activities using a coordinator and set of coordination protocols (Newcomer, Robinson, Feingold, & Jeyaraman, 2009). This framework enables participants to reach consistent agreement on the outcome of distributed activities. The coordination protocols defined in this framework accommodate a wide variety of activities, including protocols for simple short-lived operations and protocols for complex long-running business activities. For example, WS-AT and WS-BusinessActivity (WS-BA) specifications use and build upon this specification.

This specification describes a framework (see figure 3 (Newcomer, Robinson, Feingold, et al., 2009)) for a coordination service (or coordinator) which consists of three component services: (1) an activation service with an operation that enables an application to create a

coordination instance or context, (2) a registration service with an operation that enables an application to register for coordination protocols, and (3) a coordination type-specific set of coordination protocols.

Image redacted, paper copy available upon request to home institution.

**Figure 3. WS-Coordination Specification**

## 2.5 WS-Business Activity

Web Service Business-Activity (WS-BA) specification supports long-running conversations in B2B applications where the locking of resources for a longer duration is practically not feasible (Newcomer, et al., 2007). In WS-BA architecture, services are requested to perform an operation. During the performance of an operation, if there is a need to revert the changes, the business activity will cancel the operation and will inform the service to undo the changes ((Little, et al., 2004), pp.328).

## 2.6 JBoss Transactions

JBoss Transactions (JBossTS) is the premier open source transaction manager technology used in the industry for the past 20 years. It is compatible with various standards including OMG and Web Services transactions (JBossTS, 2014).

Here are some of the salient features of JBoss Transactions (JBossTS, 2014):

### **Bullet Proof Reliability:**

JBoss Transactions has evolved on industry proven technology over 20 years as a leader in the field of transaction processing.

### **Reduced Operating Costs:**

JBoss Transactions has built-in failure recovery components that can handle failures automatically with no manual intervention required. The product can be downloaded free of cost as it is an open source.

### **Flexible Deployment Options:**

JBoss Transactions can be deployed JBoss Application Server as a stand-alone as well as within a range of different container implementations.

### **Simplicity...Not Complexity:**

JBoss Transactions simplifies application development as programmers are able to focus on business logic rather than specialist non-reusable, error-prone failure recovery code.

**Full Distributed Transactions:**

JBoss Transactions preserves overall system state integrity regardless of the topology of the deployment and creates a unified transaction solution for all resources - databases, message queues, and arbitrary custom components.

**Industry Leading Interoperability:**

JBoss Transactions extends beyond J2EE to Web Services through support of specifications like WS-Coordination, WS-AtomicTransaction and WS-BusinessActivity.

**Professional Support:**

JBoss Inc. delivers the professional support, consulting, and training that you need whether you are testing a proof of concept, deploying a mission-critical application, or rolling out JEMS across your enterprise. JBoss is also continuing to partner with Arjuna Technologies.

## 2.7 Related Work

Congiu et.al. (Congiu, Grawinkel, Narasimhamurthy, & Brinkmann, 2012) demonstrate how 1PC can be considered as a low overhead atomic commitment protocol for scalable metadata services. The increase in the number of client machines in a computing cluster infrastructure makes it difficult to handle the incoming requests by using a centralized metadata server. This poses a problem to manage distributed transactions such as CREATE, DELETE and RENAME. The existing 2PC protocol to handle distributed transactions is very expensive as there are a significant number of message exchanges

between metadata servers (participants) and also synchronous writing of logs to a data store to keep the important information. In addition, this protocol locks the resources until it logs the information. Thus, simultaneous operations on same directory will not be possible and it makes the request to those resources in a serialized manner. This will have a significant performance hit when the number of requests to create new files in the same directory is very high. The solution proposed here to handle distributed transactions using and guaranteeing atomicity is to use 1PC protocol with some customization. The proposed 1PC mechanism was evaluated by comparing its performance against other protocols using ACID Sim Tools simulation framework. To aid the comparison, the following operations were performed: synchronous and asynchronous log writes and message exchanges both including for critical path. The mechanism was assessed using computational latency, network latency, and disk bandwidth. Authors state that 1PC can comparatively gain more than 55% performance based on their analysis. This mechanism is optimization in distributed file system whereas this thesis focuses optimization in transactions in a distributed web services environment with WS-Atomic Transaction specification.

Al-Houmaily and Chrysanthis (Al-Houmaily & Chrysanthis, 2004) proposed a new protocol called One-Phase, Two-Phase Commit (1-2PC) protocol which can be used to maintain the atomicity and commit the transactions in a distributed Wide Area Network amongst different web applications. This protocol dynamically selects between the two protocols depending upon the need. The mechanism shows significant performance improvement for 1PC protocol while it still maintains the characteristics of two-phase commit protocol. This protocol accommodates both One-Phase and Two-Phase commit

protocols despite of their incompatibilities and achieves the successful commitment of transactions in a distributed database environment. 1-2PC initializes with 1PC and switches to 2PC if required. Similar to other protocols, a coordinator keeps track of information in a protocol table such as the identity of each participant that takes part in the execution of the transaction. It also keeps track of the protocol used such as 1PC or 2PC. As mentioned earlier, each transaction starts with 1PC. Each participant also keeps track of active coordinators which is known as recovery-coordinators' list (RCL). RCL is kept in a stable log and is used during the post failure recovery of the participant. For optimized searching of active coordinator in the RCL, all-active flag (AAF) is set for active coordinators. Upon successful execution, the participant sends ACK message to the coordinator and in case of failure, it sends a NACK. For an update operation, if all consistency constraint validation is maintained, it follows 1PC and enters the prepared-to-commit state to invoke the final decision. If the update operation experiences deferred validation of consistency constraint, the participant notifies the coordinator and switches to 2PC by setting up an unsolicited deferred consistency constraint (UDCC) flag as a part of ACK. In this case, the participant does not enter a prepared-to-commit state as the decision depends on the message sent from coordinator. Moreover, 1-2PC protocol can handle both communication failure and site failures. So this protocol is a perfect fit in environments which have a high volume of short transactions. The protocol is evaluated by comparing 1-2PC protocol with different protocols. The basis of comparison was for commit and abort cases on a per transaction basis. In both cases protocol was assessed by considering following factors: log force delays, total forced log writes, message delays (commits and locks), total messages, and total messages with piggybacking. It has been observed that the overall overhead in 1-2PC

protocol is comparatively less and it was proven by comparing the performance of different protocols analytically with respect to log, message and time complexities. This mechanism of 1-2PC protocol dynamically makes a selection between 1PC and 2PC based on the situation in a distributed WAN environment, whereas this thesis will be focusing on designing a mechanism to enable 1PC capability in a distributed web service environment which uses WS-Atomic Transaction specification.

Neto and Reverbel (Neto & Reverbel, 2008) report the lessons learned from designing and implementing WS-Coordination and WS-AtomicTransaction. One of the reasons to use web services is the interoperability in the heterogeneous and secured environment. But this is not enough as it does not handle data inconsistency issue in enterprise application paradigm. To handle data inconsistencies, there is a need to provide transactional support. WS standards like WS-Coordination (WS-C) and WS-AtomicTransaction (WS-AT) have provided transactional support. These services were used to build a custom service which was implemented as a plugin to XActor (a distributed transaction manager that supports an open-ended set of transports, and enhances it with full support for atomic transactions over Web services). This plugin extends the capability of XActor to provide full support for atomic transactions over Web services, including crash recovery capabilities. It cooperates with XActor to transparently handle all the complex interactions that take place between the participants involved in a distributed transaction.

The WS-C/WS-AT service is built upon XActor. This service was implemented as a third party TRMI (Transaction Remote Method Invocation) plugin which is used to encapsulate the SOAP/HTTP invocation mechanism. This plugin comprises of:

- Web services that implement the WS-C/WS-AT port types
- Interceptors to propagate and import the transaction context,
- A software layer that encapsulates the SOAP/HTTP invocation mechanism and makes it available to XActor through well-known interfaces, and
- Another software layer that extends the crash recovery mechanism of XActor.

Neto and Reverbel described the design and implementation of a plug-in that enhances XActor with atomic transactional support in Web services. The concluding part discusses the lessons learned during the design and implementation of WS-C and WS-AT:

- WS-AT should support 1PC
- WS-AT should address heuristic management
- WS-AT should standardize Xid URIs
- Transaction managers should be extensible
- Dependence on a SOAP stack is burdensome i.e. tight coupling
- Performance-critical Web services require lightweight technologies.

Neto and Reverbel listed the problems they encountered during the design and implementation of WS-C and WS-AT. The problems encountered by them were a major reason to define the research problem of this thesis.



## Chapter 3

### 1PC FOR WS-AT

The objective of this thesis is to design a mechanism which can support 1PC in WS-AT for single participant transaction scenarios. Short-lived transactions involving only one participant can commit without requiring the initial “prepare” phase. Thus, there is no overhead to check whether the participant is prepared to either commit or rollback.

1PC is an optimization selected by the coordinator when it observes that only one participant has been registered for the transaction. In many cases the participant will not know that it is alone. The participant should notify the coordinator that it is capable of participating in a 1PC or 2PC protocol. The coordinator would then select the 1PC optimization if there is only one participant registered for the transaction and only if that participant is 1PC capable.

WS-AT allows individual participants to register for different protocols. There is a need to add another one of those protocols for a 2PC protocol that can also do 1PC. Thus, the proposed solution in this research is to develop a 1PC protocol mechanism that can work along with WS-AT.

### 3.1 Architecture Diagram of the Proposed Solution

Figure 4 shows the architecture diagram of the proposed solution to enable 1PC in WS-AT.

Here is how it works:

Step 1. The client application (CA) is the initiator of the transaction. CA will send a request to the Transaction Coordinator (TC) which supports the WS-AT transaction model.

Step 2. The Transaction Coordinator will send a response with the transaction context to the client application. The response also contains the endpoint reference to register the web service where participants are enlisted.

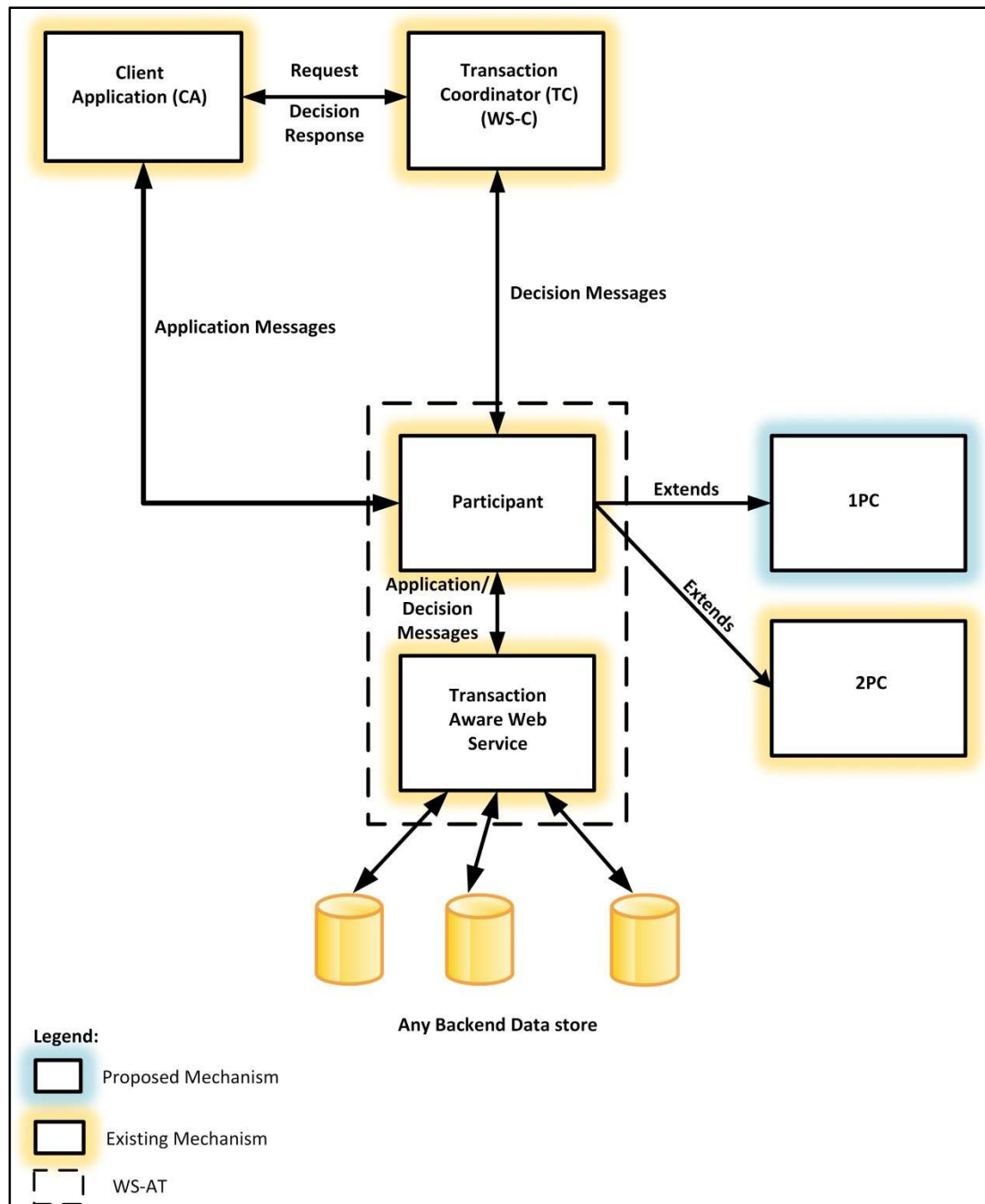
Step 3. The participant of transaction aware web services currently extends the 2PC protocol. For more information on 2PC, please refer to Chapter 2. With the new mechanism, the participant now extends 1PC which helps to enable 1PC in WS-Atomic transaction aware web services.

Step 4. The participant will inform the Transaction Coordinator about its capability to handle 1PC. The coordinator will monitor the number of participants enlisted in the atomic transaction and based on that invoke either 2PC or 1PC as an optimization.

Step 5. The client now registers itself as a participant and communicates with WS-Atomic transaction aware web service to execute the business logic required to complete the desired task.

Step 6. The result will be a “successful completion of the task” or a “failure”. In case of 1PC success the participant instead of sending the decision response to the

Transaction Coordinator, will directly send the response as commit or rollback to the client application in case of failure.



**Figure 4. Architecture Diagram of the Proposed Mechanism**

### 3.2 Selection of Implementation Platform

From the technical implementation point of view, there are different tools and platforms such as IBM WebSphere Application Server, JBoss Application Server, .Net, Oracle WebLogic Server, etc. available in the market that support WS-Atomic Transaction. Research was done on the feasibility and flexibility to implement the proposed mechanism. Technical documents of IBM WebSphere Application Server and JBoss Application Server were referred before making the decision.

IBM WebSphere Application server has extensive support for WS-Atomic Transaction. But it needs licensing in order to use since it is commercial. It is widely used in the industry. It requires a machine with a high-end hardware configuration. Therefore, from a cost benefit analysis view point, it is not feasible to implement the proposed solutions for this research using IBM.

JBoss is an open source platform with the General Public License (GPL), and hence using this platform will meet all conditions of economic feasibility. The JBoss technical team has a deep interest in this research. JBoss has an active technical forum which provides a development guide using the JBoss Transaction API which will be used in this research. Detailed technical documentation is also available for the JBoss Application server, transaction API, and integration with the platform. The support for WS-AT specification in JBoss is extensive. Therefore, integration of IPC mechanism on top of the existing implementation would not be a problem. The research would be completed in a specified

time frame which can be considered as feasible as far as scheduling is concerned. Moreover, the hardware configuration required for JBoss platform is not comparatively high and hence, the implementation can be done with a machine having the average hardware configuration. Considering all these aspects, it is feasible from a technical view point. As mentioned earlier, the JBoss technical forum is an excellent source of information and knowledge base which will be helpful during the development of the proposed mechanism. Therefore, the decision was made to use the JBoss platform for the technical implementation of the proposed mechanism.

Here are the high-level details of the software and hardware requirements:

- Java SE 6.x or 7.x
- JBoss WildFly 8.x (Application Server)
- JBoss web service transaction API
- Eclipse as IDE (Integrated Development Environment)

### 3.3 Implementation of IPC Mechanism

Technical implementation to validate this research has been completed successfully. The tools used are:

- Java SE 6.x or 7.x
- JBoss WildFly 8.x (Application Server)
- JBoss XTS (XML Transaction Service) API
- Eclipse as IDE (Integrated Development Environment)

The JBoss XTS currently supports protocols like WS-Coordination (WS-C), WS-Atomic Transaction (WS-AT) and WS-Business Activity (WS-BA) (JBossTS, 2014). In this research, WS-C and WS-AT are used. The existing API supports classes for participants for 2PC transaction. Figure 5 shows the sequence diagram of 2PC mechanism. The implementation of a normal 2PC scenario is as follows:

- The user will send a request to the Servlet, which in turn, will initiate the transaction with a User Transaction object.
- Next the servlet class will invoke an implementation class with business logic.
- This class will initiate Transaction Coordinator, which in turn, enlists 2PC supported Participant class which is WS-AT transaction aware web service and extends the main Participant class.
- During the Transaction Coordinator initiation, it will register and activate the participants via Participant Processor.
- Participant Processor will now activate the ParticipantStub for 2PC, which in turn activates the Coordinator Engine. Coordinator Engine maintains the state of the Transaction Coordinator.
- Since this is 2PC, the ParticipantStub for 2PC will initiate the Coordinator Engine with “Active” state of WS-AT as an initial state.
- The Participant Engine maintains the state of Participants.
- The Transaction Coordinator activates the Participant Engine with the initial state set as “Active”. This will trigger the “prepare” phase.

- The Transaction Coordinator will wait for the votes from all the registered participants.
- During this phase, the 2PC supported Participant class goes through the “Preparing”, “Prepared” and “PreparedSuccess” states of WS-AT.
- It will send a response back to the Transaction Coordinator and changes the state to “Committing”. This is a “commit” phase. Finally the Transaction Coordinator will make the decision based on the response and will successfully commit transaction via Coordinator Engine.

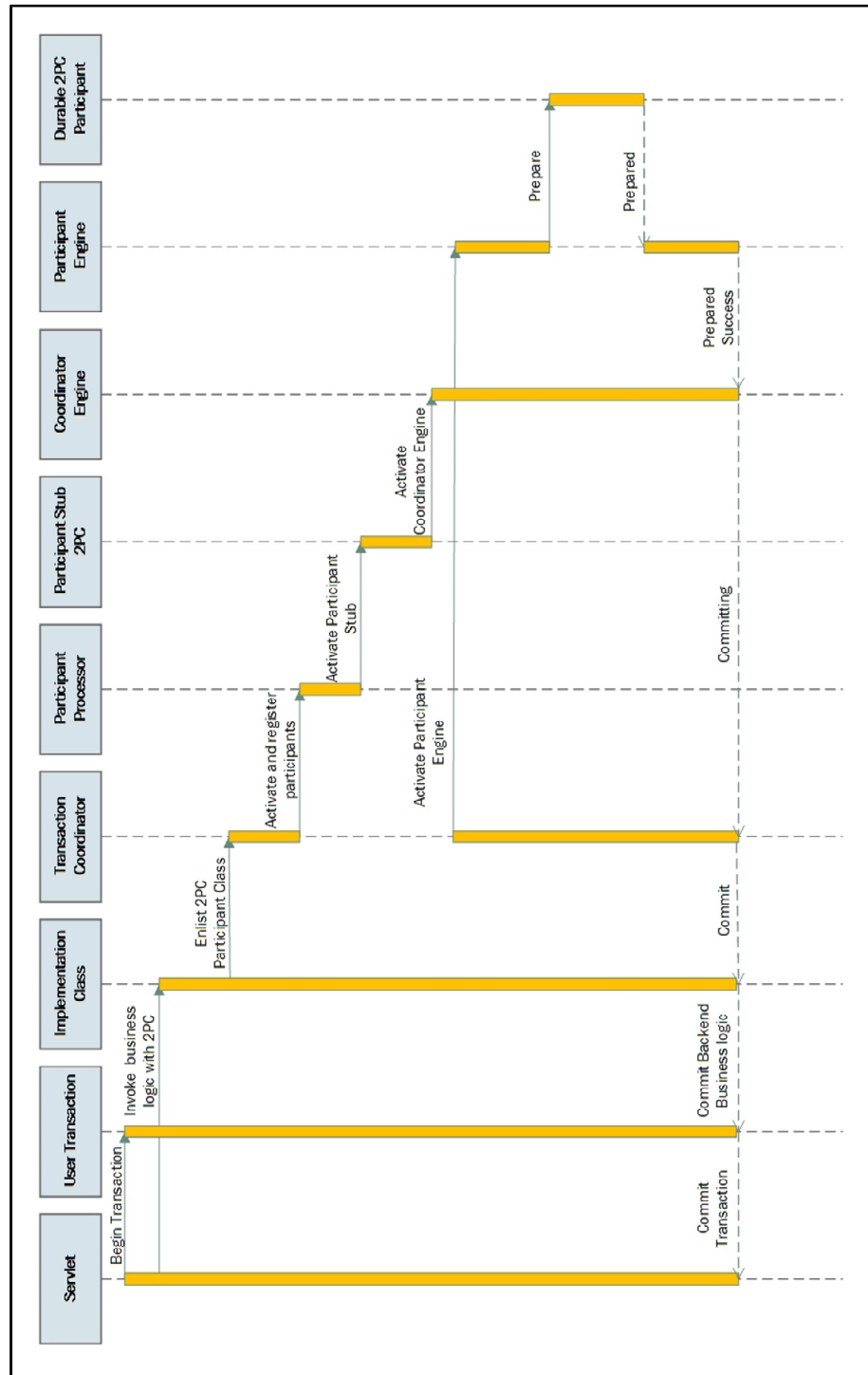
In this research, the support was added for participants in 1PC transaction. Figure 6 shows the sequence diagram of 1PC mechanism. This is how it is implemented:

- The user will send a request to the Servlet, which in turn, will initiate the transaction with a User Transaction object.
- Next the servlet class will invoke an implementation class with business logic.
- This class will initiate Transaction Coordinator, which in turn, enlists 1PC supported Participant class which is WS-AT transaction aware web service and extends the main Participant class. This is a new class created to achieve 1PC mechanism which is isolated from existing classes that support 2PC mechanism.
- During Transaction Coordinator initiation, it will register and activate the participant via Participant Processor.
- Participant Processor will now activate the ParticipantStub for 1PC, which in turn activates the Coordinator Engine. Coordinator Engine maintains the state of a Transaction Coordinator. The ParticipantStub for 1PC is another class created to

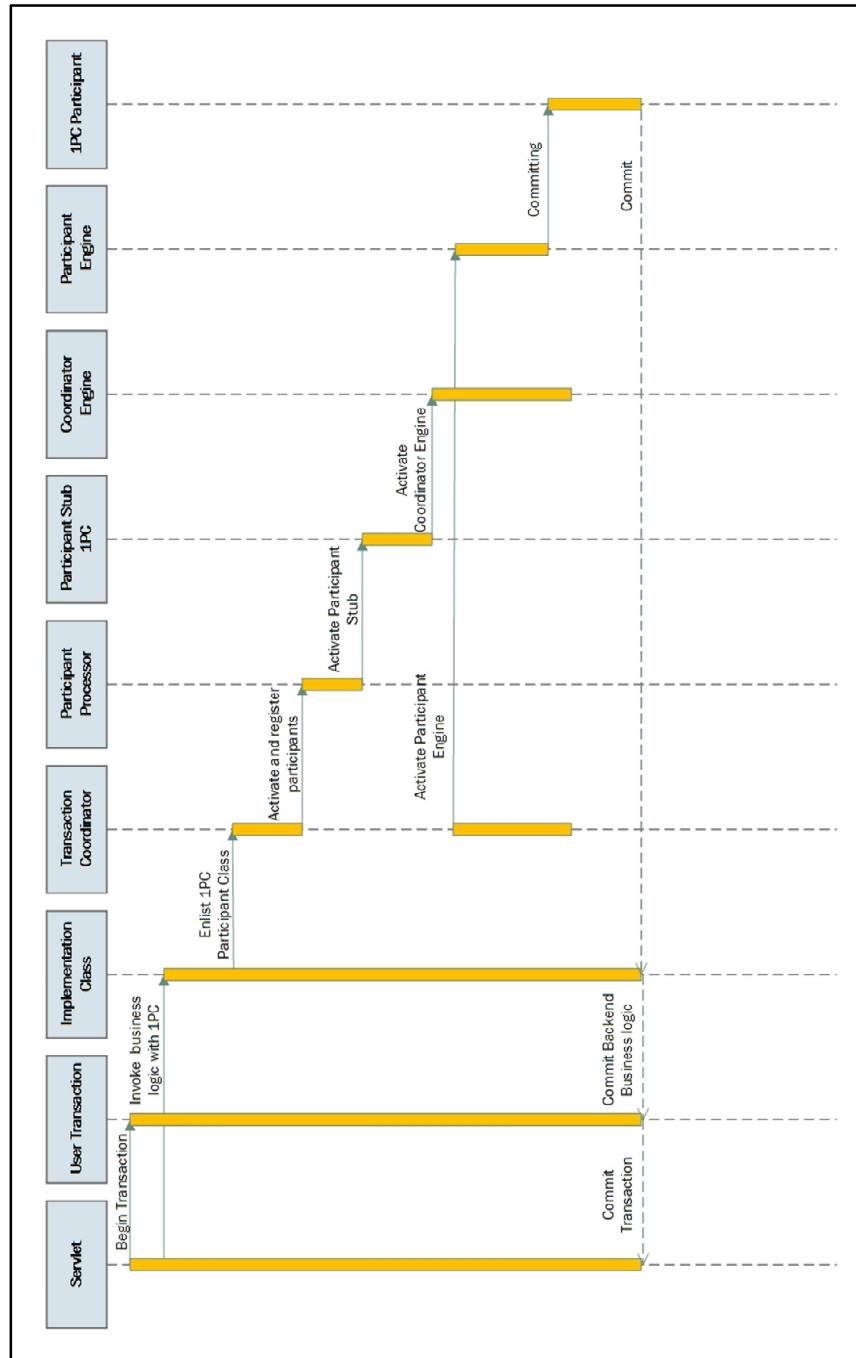
achieve 1PC mechanism in isolation with existing class supporting 2PC mechanism.

- Since this is 1PC, the Participant Stub for 1PC will initiate the Coordinator Engine with “Committing” state of WS-AT as an initial state.
- The Participant Engine maintains the state of Participants.
- Transaction Coordinator activates the Participant Engine with the initial state set as “Committing”.
- The Participant Class which supports 1PC, will not execute the “Prepare” phase as the initial state is “Committing” unlike 2PC scenario where the initial state is “Active”
- The participant in this case will execute “commit” phase of the transaction and successfully commits the transaction without responding back to Transaction Coordinator.





**Figure 5. Sequence Diagram for existing 2PC mechanism**



**Figure 6. Sequence Diagram for new 1PC mechanism**

## Chapter 4

### RESEARCH METHODOLOGY

The research methodology used in this thesis is Design Science Research methodology.

#### 4.1 Design Science Research Methodology

Design science research involves the design of novel or innovative artifacts and the analysis of the use and/or performance of such artifacts to improve and understand the behavior of aspects of Information Systems (IS). Such artifacts include, but certainly are not limited to, algorithms (e.g. for information retrieval), human/computer interfaces and system design methodologies or languages.

The design science research paradigm is highly relevant to information systems (IS) research because it directly addresses two of the key issues of the discipline:

- The central, despite controversial, role of the IT artifact in IS research (Benbasat & Zmud, 2003; Orlikowski & Iacono, 2001; Weber, 1987), and,
- The perceived lack of professional relevance of IS research (Benbasat & Zmud, 1999; Hirschheim & Klein, 2003).

Design science research in IS addresses what are considered to be wicked problems, like (Hevner & Chatterjee, 2010):

- Ill-defined problems with unstable requirements,
- Problems with complex set of interactions among its components,
- Problems that require flexibility to change design processes and artifacts,
- Problems that require highly creative solutions, and
- Problems that rely on human social abilities (e.g., teamwork) to produce effective solutions.

## 4.2 Design Science Research Guidelines

The design science methodology in the IS discipline has been defined via a conceptual framework for understanding information systems research. There are certain guidelines which help the researchers to conduct and evaluate their research based on this methodology. Again these are just guidelines and not strict enforcement of laws (Hevner & Chatterjee, 2010).

### **Guideline 1 – Design as an Artifact:**

As per the first guideline, a design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation. The focus of this thesis is to create a conceptual mechanism for WS-AT with IPC. Preliminary details of this proposed mechanism is described in Chapter 3. Thus, this thesis follows the design as an artifact guideline by designing a IPC mechanism for WS-AT.

**Guideline 2 – Problem Relevance:**

As per the second guideline, the objective of a design science research should be development of technology-based solutions to important and relevant business problems. 1PC can reduce overhead costs during short-lived transaction scenario involving one participant. The relevance of this problem is established in Chapter 1. This thesis follows the problem relevance guideline as well.

**Guideline 3 – Design Evaluation:**

As per the third guideline, the utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods. This thesis follows the design evaluation guideline as the 1PC mechanism will be implemented by developing a proof of concept (POC) system and evaluated using three different scenarios as discussed in Chapter 5.

**Guideline 4 – Research Contributions:**

As per the fourth guideline, an effective design science research should provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies. This thesis makes research contributions on several fronts. It produces a design artifact by developing a conceptual mechanism for WS-AT with 1PC. It develops a proof of concept system which implements the 1PC mechanism using JBoss Transaction, and subsequently evaluates the proposed mechanism to demonstrate its utility.

**Guideline 5 – Research Rigor:**

As per the fifth guideline, a design science research should rely upon the application of rigorous methods in both the construction and evaluation of the design artifact. This thesis utilizes appropriate methods in the construction and evaluation of the 1PC mechanism. The 1PC mechanism is built using the JBoss Transaction API and subsequently evaluated in terms of reduction in overhead due to a fewer number of message exchanges and hence improvement of the performance in terms of execution time.

**Guideline 6 – Design as a Search Process:**

As per the sixth guideline, search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment. During the process of designing and developing the 1PC mechanism and the POC system, appropriate alternatives were considered and alternatives with appropriate merit and fit to the problem were selected.

**Guideline 7 – Communication of Research:**

As per the seventh guideline, a design science research must be presented effectively to both technology-oriented and management-oriented audiences. This thesis follows communication guideline as the research work is communicated in the form of a written document and an oral presentation as a part of a thesis proposal and a final defense.

## Chapter 5

### EVALUATION

WS-Atomic Transaction (WS-AT) specification currently supports Two-Phase Commit (2PC) protocol for the short-lived transaction scenario. In many situations, 2PC is not required. Instead, One-Phase Commit (1PC) can suffice and produce the similar desired results as 2PC with optimization. The problem is that WS-AT does not currently support 1PC which makes the short-lived transaction very expensive in terms of overhead costs, Transaction Coordinator failure, and wait time in decision making. The ideal scenario in a distributed service environment where 1PC can be used instead of 2PC when there is only a single participant. This research focuses on solving the above real world problem. The solution is evaluated both statistically and practically.

The 1PC protocol mechanism implemented in this research is tailored for WS Atomic Transaction specification with one participant scenario. The 1PC mechanism is an optimization that is expected to reduce the time taken to complete transactions in one participant scenarios. In the case of a single participant in a distributed environment, enabling one phase protocol can reduce the overhead by cutting down the “prepare” phase as compared to 2PC. This will definitely improve the performance in terms of execution time (a.k.a., response time). The reason behind this is that there are fewer messages that need to be exchanged between the Transaction Coordinator and Participant in a 1PC scenario.

Another important factor that can significantly improve the performance is there are fewer recovery logs to be written as the outcome of the transaction is solely based on the decision made by the single participant. Transaction Manager does not take part in the decision-making process in a 1PC scenario. In a typical I/O operation, writing recovery logs are more expensive as compared to writing data on a disk because all the resources will be blocked until the disk confirms that the logs have been written successfully.

By avoiding the “prepare” phase, the participant will also show a significant performance improvement as it is not blocked by the coordinator between “prepare” and “commit” phases, which is not the case in a typical Two-Phase Commit scenario. In 2PC, a prepared Participant has to wait until it gets the final decision message from the Transaction coordinator before it can proceed. In the case of Transaction Coordinator failure, this notification can be delayed or may not be received in the case of failure at the Transaction Coordinator level. This would not be the case with the 1PC mechanism since the participant does not have to wait for a decision from Transaction Coordinator.

Thus, the objective of evaluation is to test whether 1PC mechanism performs better than 2PC mechanism in terms of execution time (a.k.a., response time). 1PC and 2PC mechanisms are evaluated to test their performance in a simulated environment using business process scenarios with single participant.

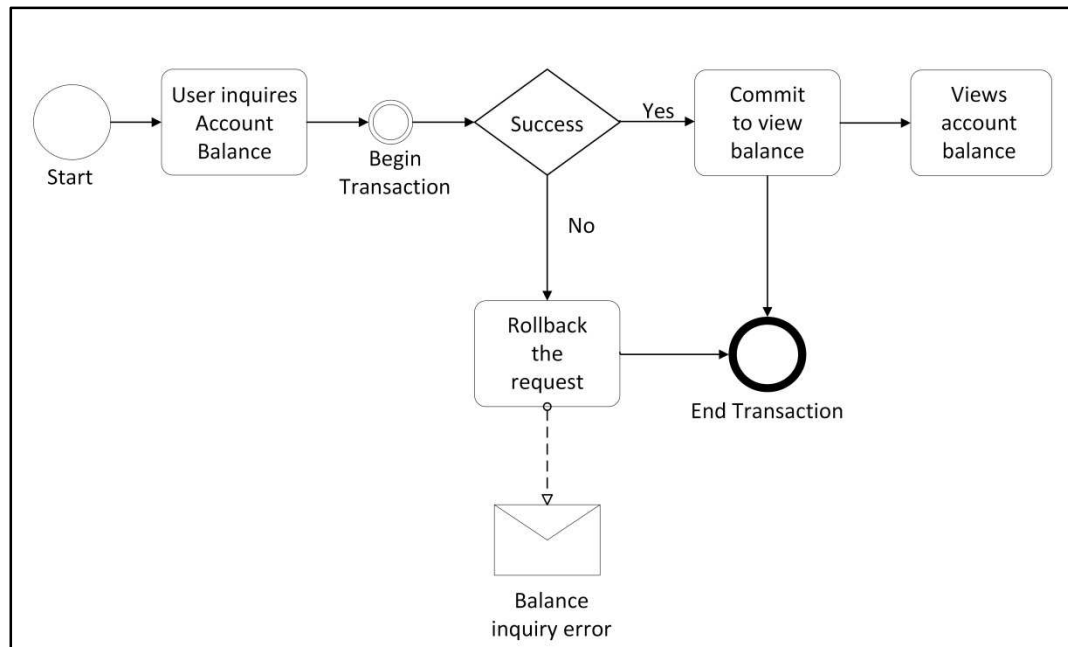


## 5.1 Experimental Setup Scenarios

Single participant business process scenarios form the experiment setup developed to evaluate 1PC and 2PC mechanisms for WS-AT. We have developed three business process scenarios that best fit with this research as they utilize web service and include a single participant. The aim is to implement identified scenarios using both 1PC and 2PC mechanisms, and independently gather performance data, i.e., execution time.

### 5.1.1 Scenario 1: Balance Inquiry in Online Banking

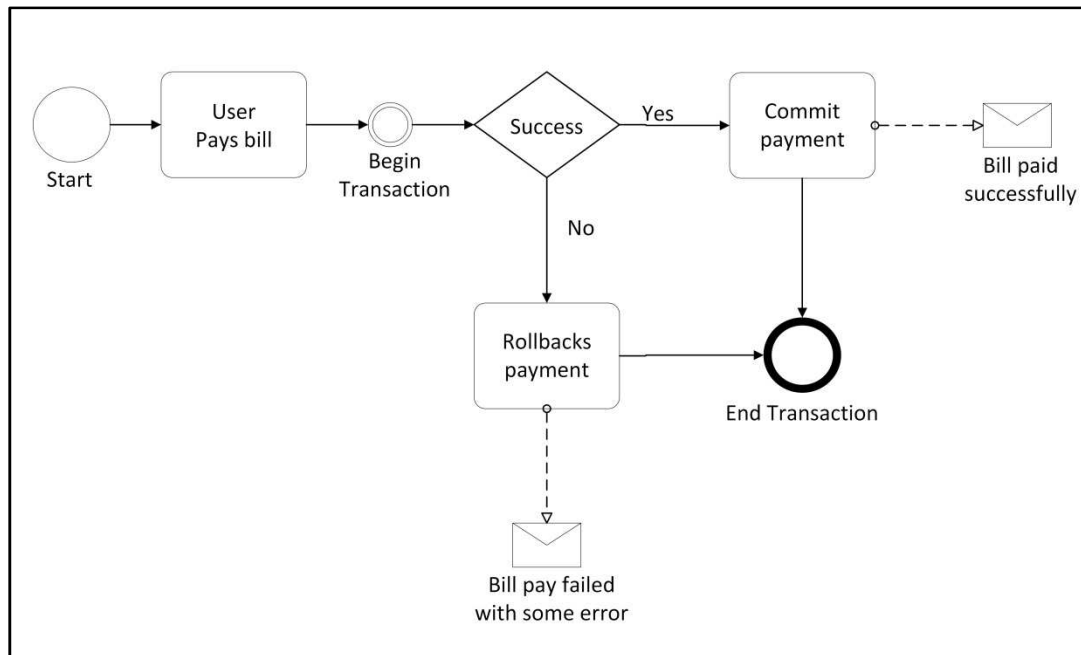
In this scenario, the user does an inquiry on account balance. The participant here will be called as “balanceinquiry”. Since there is a single participant in this scenario, 2PC is not required. The web service will simulate the process of user inquiring on his account balance. Figure 7 shows the Business Process Model and Notation (BPMN) diagram for the restaurant table booking process. BPMN diagram is the industry standard to depict business processes (BPMN, 2011).



**Figure 7. BPMN Diagram for Inquiring Bank Account Balance process**

#### 5.1.2 Scenario 2: Bill Pay

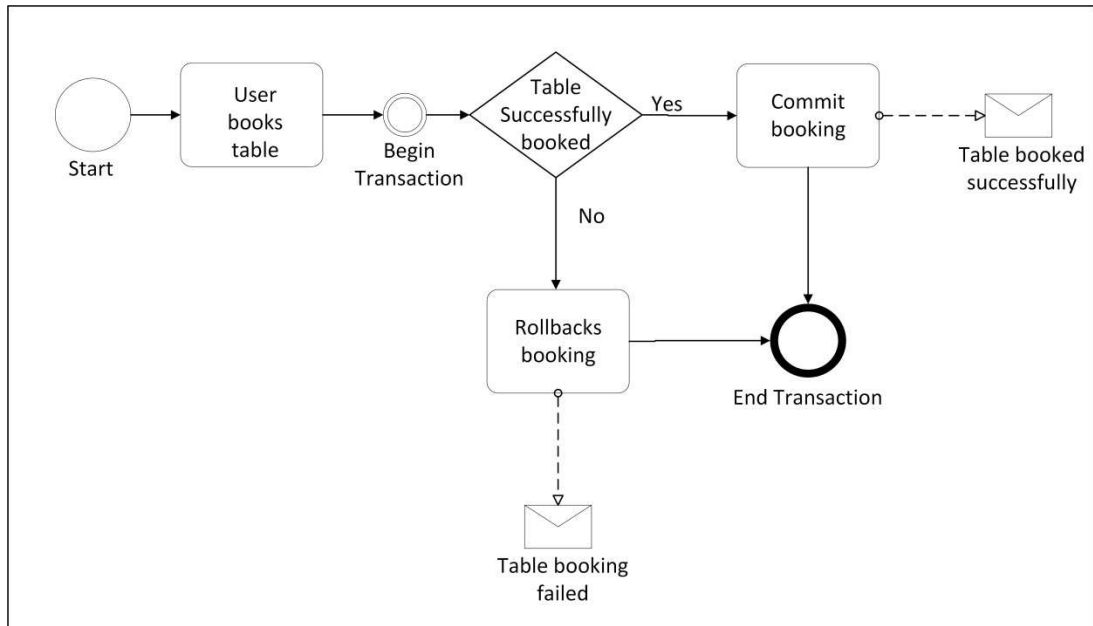
In this scenario, the user pays a bill. This scenario has only a single participant called “paybill”. Therefore, 2PC is not needed in this scenario. The web service simulates the process of paying a bill where the user can pay bill. Figure 8 shows the BPMN diagram for the bill paying process.



**Figure 8. BPMN Diagram for Bill Paying process**

### 5.1.3 Scenario 3: Restaurant Table Booking

In this scenario, the user books a table in a restaurant for a family. This scenario has only a single participant called “makebooking”. Therefore, 2PC is not needed in this scenario. The web service simulates the process of booking a table in a restaurant. Figure 9 shows the BPMN diagram for the restaurant table booking process.



**Figure 9. BPMN Diagram for Restaurant Table Booking process**

## Chapter 6

### EVALUATION RESULTS

#### 6.1 Independent Samples t-Test

In this research, we aim to analyze the statistical difference between data sets gathered from 1PC and 2PC implementations for three experimental scenarios. For this analysis, we have two unrelated or independent groups (1PC and 2PC mechanisms) and a dependent variable (execution time, a.k.a., response time). Independent Samples t-Test is appropriate for this research as it complies with the following assumptions ((Morgan, Leech, Gloeckner, & Barrett, 2007), pp.143-144):

- **Assumption 1:** Both groups should have equal variances of the dependent variable. Levene's test for equal variances is used to see if Assumption 1 is met or not. If the Levene's test is not significant (i.e.  $p > 0.05$ ), then the assumption is not violated, and data from "Equal variances assumed" row can be used for interpreting the t-Test and related statistics. If the Levene's test is significant (i.e.  $p < 0.05$ ), then data from "Equal variances not assumed" row should be used, as it violates the assumption of equal variances.
- **Assumption 2:** Within each group, the dependent variable is normally distributed. Independent Samples t-Test is robust to that extent that even if this assumption is violated the results can still be considered as normal. Therefore, this assumption is not taken into consideration when the independent samples t-Test is performed.

- **Assumption 3:** The data for both groups are independent. To meet this assumption, the data collection should be such that the groups do not interfere with each other. As the 1PC mechanism implementation is independent of the 2PC mechanism implementation, data collected for the 1PC mechanism is independent and unrelated to that of the 2PC mechanism.

IBM SPSS statistics version 22 (SPSS, 2014) is used to perform the t-Test for independent samples. As discussed in the Chapter 5, the experimental data set was gathered from simulated study of different business process scenarios:

Scenario 1: Online Bank Balance Inquiry

Scenario 2: Online Bill Pay

Scenario 3: Restaurant Table Booking

For each scenario, this research measures the execution time (a.k.a., response time) of 1PC and 2PC mechanisms for 50 users, 100 users, 150 users and 200 users. Data for both groups are independent, i.e. the execution time of 1PC is independent and does not interfere with that of 2PC. A total of twelve data sets were collected for three scenarios with four variations each. Detailed explanation of t-Test results has been provided for Scenario 1 (Online Balance Inquiry) with 50 users. Similar explanation holds true for 100 users, 150 users, and 200 users for Scenario 2 and 3. Table 1 shows descriptive statistics for two independent groups, 1PC and 2 PC, for a population of 50, 100, 150, and users groups for Scenario 1.

Table 2 shows a summary of independent samples t-Test results for Scenario 1 with a population of 50 users. Levene's test is for Assumption number 1, i.e., variances of the two groups is equal. It can be observed that Levene's test is not significant ( $p > 0.05$ ); therefore, data from "Equal variances assumed" row is used for analysis. From Table 2, it can also be observed in the "t-Test for Equality of Means" column, at the 5% level of significance there is significant (i.e.,  $p < 0.05$ ) difference in execution time between 1PC and 2PC groups.

From Table 1, it is inferred that 1PC had a lower mean for execution time in comparison to the 2PC mechanism. Thus, the execution time for 1PC mechanism is significantly different from the execution time for 2PC mechanism. The column "95% Confidence Interval" indicates that if the experiment is repeated 100 times, then 95 times the difference will fall within the confidence interval (upper bound and lower bound). For other scenarios, the same explanation on the confidence interval holds true.

Table 3 shows a summary of the independent samples t-Test results for Scenario 1 with a population of 100 users. It can be observed that Levene's test is not significant ( $p > 0.05$ ); therefore, data from "Equal variances assumed" row is used for analysis. It can be also be observed that in the "t-Test for Equality of Means" column, at the 5% level of significance there is significant (i.e.,  $p < 0.05$ ) difference in execution time between 1PC and 2PC groups. Similar explanation holds true for Scenario 1 with 150 users. From table 4, it can be observed that there is a significant difference between 1PC and 2PC groups for Scenario 1 with 150 users.

No. of Users	Protocol Mechanism	Sample Size (N)	Mean	Std. Deviation	Std. Error Mean
50	1PC	50	488.880	298.011	42.145
	2PC	50	806.620	377.224	53.348
100	1PC	100	413.890	201.854	20.185
	2PC	100	588.340	239.990	23.999
150	1PC	150	410.780	273.655	22.344
	2PC	150	572.780	286.462	23.390
200	1PC	200	379.810	180.758	12.782
	2PC	200	547.030	246.932	17.461

**Table 1. Group Statistics for Scenario 1 - Online Balance Inquiry**

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	0.366	0.547	-4.674	98	0.000	-317.740	67.987	-452.657	-452.748
Equal variances not assumed			-4.674	93.018	0.000	-317.740	67.987	-182.823	-182.732

**Table 2. Independent Samples t-Test Results for Scenario 1 with 50 Users**



Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	3.406	0.066	-5.563	198	0.000	-174.450	31.359	-236.291	-112.609
Equal variances not assumed			-5.563	192.352	0.000	-174.450	31.359	-236.302	-112.598

**Table 3. Independent Samples t-Test Results for Scenario 1 with 100 Users**

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	0.337	0.562	-5.008	298	0.000	-162.000	32.347	-225.657	-98.343
Equal variances not assumed			-5.008	297.379	0.000	-162.000	32.347	-225.658	-98.342

**Table 4. Independent Samples t-Test Results for Scenario 1 with 150 Users**

Table 5 shows a summary of independent samples t-Test results for Scenario 1 with a population of 200 users. It can be observed that Levene's test is significant ( $p < 0.05$ ); therefore, data from "Equal variances not assumed" row is used for analysis. It can be also be observed that in the "t-Test for Equality of Means" column, at the 5% level of significance there is a significant (i.e.,  $p < 0.05$ ) difference in execution time between 1PC and 2PC groups.

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	10.369	0.001	-7.728	398	0.000	-167.220	21.639	-209.761	-124.680
Equal variances not assumed			-7.728	364.691	0.000	-167.220	21.639	-209.773	-124.667

**Table 5. Independent Samples t-Test Results for Scenario 1 with 200 Users**

Table 6 shows descriptive statistics for two independent groups 1PC and 2 PC for a population of 50, 100, 150, and 200 users groups for Scenario 2 (Online Bill Pay). Table 7 shows a summary of independent samples t-Test results for Scenario 2 with a population of 50 users. It can be observed that Levene's test is not significant ( $p > 0.05$ ); therefore, data from "Equal variances assumed" row is used for analysis. It can be also be observed that in the "t-Test for Equality of Means" column, at the 5% level of significance there is significant (i.e.,  $p < 0.05$ ) difference in execution time between 1PC and 2PC groups. Similar explanation holds true for Scenario 2 with 150 users. From Table 9, it can be observed that there is a significant difference between 1PC and 2PC groups for Scenario 2 with 150 users.

No. of Users	Protocol Mechanism	Sample Size (N)	Mean	Std. Deviation	Std. Error Mean
50	1PC	50	536.180	376.069	53.184
	2PC	50	757.340	428.335	60.576
100	1PC	100	336.940	220.957	22.096
	2PC	100	565.770	331.307	33.131
150	1PC	150	379.473	253.344	20.685
	2PC	150	494.040	291.751	23.821
200	1PC	200	310.335	197.316	13.952
	2PC	200	477.3850	241.61063	17.08445

**Table 6. Group Statistics for Scenario 2 - Online Bill Pay**

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	0.130	0.719	-2.744	98	0.007	-221.160	80.610	-381.128	-61.192
Equal variances not assumed			-2.744	96.386	0.007	-221.160	80.610	-381.161	-61.159

**Table 7. Independent Samples t-Test Results for Scenario 2 with 50 Users**

Table 8 shows a summary of independent samples t-Test results for Scenario 2 with a population of 100 users. It can be observed that Levene's test is significant ( $p < 0.05$ ) therefore data from "Equal variances not assumed" row is used for analysis. It can be also be observed that in the "t-Test for Equality of Means" column, at the 5% level of significance there is significant (i.e.,  $p < 0.05$ ) difference in execution time between 1PC

and 2PC groups. Similar explanation holds true for Scenario 2 with 200 users. From Table 10, it can be observed that there is a significant difference between 1PC and 2PC groups for Scenario 2 with 200 users.

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	8.925	0.003	-5.746	198	0.000	-228.830	39.823	-307.362	-150.299
Equal variances not assumed			-5.746	172.523	0.000	-228.830	39.823	-307.433	-150.227

**Table 8. Independent Samples t-Test Results for Scenario 2 with 100 Users**

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	2.050	0.153	-3.631	298	0.000	-114.567	31.549	-176.654	-52.480
Equal variances not assumed			-3.631	292.253	0.000	-114.567	31.549	-176.659	-52.474

**Table 9. Independent Samples t-Test Results for Scenario 2 with 150 Users**

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	16.796	0.000	-7.573	398	0.000	-167.050	22.058	-210.414	-123.686
Equal variances not assumed			-7.573	382.722	0.000	-167.050	22.058	-210.420	-123.680

**Table 10. Independent Samples t-Test Results for Scenario 2 with 200 Users**

Table 11 shows descriptive statistics for two independent groups 1PC and 2 PC for a population 50, 100, 150, and 200 users groups for Scenario 3 (Online Restaurant Table Booking). Table 12 shows a summary of independent samples t-Test results for Scenario 3 with a population of 50 users. It can be observed that Levene's test is not significant ( $p > 0.05$ ); therefore, data from "Equal variances assumed" row is used for analysis. It can also be observed that in the "t-Test for Equality of Means" column, at the 5% level of significance there is a significant (i.e.,  $p < 0.05$ ) difference in execution time between 1PC and 2PC groups. Similar explanation holds true for Scenario 3 with 100 users and 200 users. From Table 13, it can be observed that there is a significant difference between 1PC and 2PC groups for Scenario 3 with 100 users. From table 15, it can be observed that there is a significant difference between 1PC and 2PC groups for Scenario 3 with 200 users.

No. of Users	Protocol Mechanism	Sample Size (N)	Mean	Std. Deviation	Std. Error Mean
50	1PC	50	457.360	360.626	51.000
	2PC	50	612.180	279.648	39.548
100	1PC	100	395.470	251.381	25.138
	2PC	100	545.220	229.262	22.926
150	1PC	150	445.627	205.949	16.816
	2PC	150	571.167	324.997	26.536
200	1PC	200	406.580	192.655	13.623
	2PC	200	510.680	205.795	14.552

**Table 11. Group Statistics for Scenario 3 - Online Restaurant Table Booking**

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	0.163	0.687	-2.399	98	0.018	-154.820	64.537	-282.892	-26.748
Equal variances not assumed			-2.399	92.280	0.018	-154.820	64.537	-282.992	-26.648

**Table 12. Independent Samples t-Test Results for Scenario 3 with 50 Users**

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	3.257	0.073	-4.401	198	0.000	-149.750	34.023	-216.843	-82.657
Equal variances not assumed			-4.401	196.344	0.000	-149.750	34.023	-216.847	-82.653

**Table 13. Independent Samples t-Test Results for Scenario 3 with 100 Users**

Table 14 shows summary of independent samples t-Test results for Scenario 3 with a population of 150 users. It can be observed that Levene's test is significant ( $p < 0.05$ ) therefore data from "Equal variances not assumed" row is used for analysis. It can be also be observed that in the "t-Test for Equality of Means" column, at the 5% level of significance there is significant (i.e.,  $p < 0.05$ ) difference in execution time between 1PC and 2PC groups.

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	12.439	0.000	-3.996	298	0.000	-125.540	31.415	-187.364	-63.716
Equal variances not assumed			-3.996	252.050	0.000	-125.540	31.415	-187.410	-63.670

**Table 14. Independent Samples t-Test Results for Scenario 3 with 150 Users**

Exp. groups: 1PC and 2PC	Levene's Test for Equality of Variances		t-Test for Equality of Means						
	F	Sig.	t	df	Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval	
								Lower	Upper
Equal variances assumed	0.059	0.808	-5.222	398	0.000	-104.100	19.933	-143.288	-64.912
Equal variances not assumed			-5.222	396.280	0.000	-104.100	19.933	-143.288	-64.912

**Table 15. Independent Samples t-Test Results for Scenario 3 with 200 Users**

It can be noted that all t-Tests resulted in significant differences between 1PC and 2PC mechanism groups. Further analysis of mean differences (see tables 1, 6, and 11) indicates that 1PC mechanism had lower mean for execution time in comparison to 2PC mechanism for all twelve experiments. Thus, reduction in the execution time produced due to optimization implemented in the 1PC mechanism is statistically significant.

## 6.2 Effect Size

Statistical significance is not the same as practical significance. Statistical significance indicates that difference between two groups are meaningful and not a random chance of occurrence, while practical significance indicates magnitude of the difference. A difference that is practically significant implies that the treatment utilized in the experiment has a



higher chance of creating substantial practical impact. It is not necessary that an experiment which is statistically significant is practically significant.

Effect size is used to determine whether the outcome of a research is practically significant or not. Effect size can be defined as “the strength of the relationship between the independent variable and the dependent variable” ((Morgan, et al., 2007), pp.92-93). There are many methods to calculate the effect size. This research follows Cohen’s guidelines and uses “The d family of effect size measures” (Cohen, 1988). According to this method, the value of d will help to determine whether the outcome is practically smaller or larger than typical effect i.e. the execution time (a.k.a., response time) of 1PC mechanism is practically lesser or more than it’s typical value. As per Cohen’s guidelines (see Equation 1), effect size (d) is absolute mean differences between 1PC and 2PC groups divided by the pooled standard deviation. The pooled standard deviation is the square root of the average of the squared standard deviations of 1PC and 2PC groups. Table 17 shows the effect size for all three experimental scenarios.

$$\text{Cohen's Effect Size (d)} = \frac{|\text{Mean of 1PC} - \text{Mean of 2PC}|}{\sqrt{\frac{(SD_{1PC}^2 + SD_{2PC}^2)}{2}}}, \text{ where SD is standard deviation.}$$

#### **Equation 1. Cohen’s Effect Size**

A smaller than typical effect size ( $d < 0.5$ ) was detected for Scenario 1 – 200 users, Scenario 2 – 150 users, Scenario 3 – 50 users and 150 users experiments. A typical effect size ( $0.5 \leq d < 0.8$ ) was detected for execution times for Scenario 1 - 100 users and 150

users, Scenario 2 -5 users and 200 users, Scenario 3 – 100 users and 200 users experiments.

A larger than typical effect size ( $d \geq 0.8$ ) was detected for execution times for Scenario 1 - 50 users and Scenario 2 - 100 users experiments.

Scenario	Users	Mean		Standard Deviation		Mean Diff.	Pooled Standard Deviation	Effect Size (d)	Post Hoc Power
		1PC	2PC	1PC	2PC				
1	50	488.880	806.620	298.011	377.224	-317.740	339.933	0.935	0.996
	100	413.890	588.340	201.854	239.990	-174.450	221.743	0.787	1.000
	150	410.780	572.780	273.655	286.462	-162.000	280.132	0.578	0.999
	200	379.810	547.030	180.758	246.932	-167.220	407.379	0.410	0.983
2	50	536.180	757.340	376.069	428.335	-221.160	403.050	0.549	0.776
	100	336.940	565.770	220.957	331.307	-228.830	281.591	0.813	1.000
	150	379.473	494.040	253.343	291.751	-114.567	273.223	0.419	0.951
	200	310.335	477.385	197.316	241.611	-167.050	220.578	0.757	1.000
3	50	457.360	612.180	360.626	279.648	-154.820	322.687	0.480	0.661
	100	395.470	545.220	251.381	229.262	-149.750	240.576	0.622	0.992
	150	445.627	571.167	205.949	324.997	-125.540	272.064	0.461	0.978
	200	406.580	510.680	192.655	205.795	-104.100	199.333	0.522	0.999

**Table 16. Effect Size and Post Hoc Power for all three Scenarios**

Thus, the mean difference between 1PC and 2PC mechanisms for execution time for Scenario 1 with 50, 100, and 150 users; and Scenario 2 with 50, 100, and 200 users; and Scenario 3 with 100 and 200 users experiments are of statistical significance and have moderate to high practical significance. Whereas, for the rest of scenarios, the results were statistically significant but were not of practical significance as effect sizes are lower than 0.50. In the next section, we provide discussion on post-hoc power analysis.

### 6.3 Power Analysis

Post hoc (means “after the fact”) analysis is conducted as follow-up tests after statistical analysis to further examine relationships between subgroups of sampled population ((Spatz, 2010), p. 248). Post hoc statistical power analysis was performed using G\*Power (Faul, Erdfelder, Lang, & Buchner, 2007) software to determine the likelihood of finding statistical difference between 1PC and 2PC mechanisms for the given sample size and observed effect size. Post hoc power that ranges between 0.5 and 0.8 can be considered as adequate power and Post hoc power greater than 0.8 as high power (Onwuegbuzie & Leech, 2004). Table 17 shows the post hoc power for all three experimental scenarios.

Adequate power was observed for Scenario 2 with 50 users and Scenario 3 with 50 users experiments. High power was observed in the rest of the experiments. Thus, for all the experiment scenarios utilized in this research there is a high probability of observing similar findings in future experiments with a similar structure, effect size, and standard deviation at the 5% level of significance, with the exception of Scenario 2 with 50 users and Scenario 3 with 50 users which has moderate probability.

In summary, out of twelve scenarios, all scenarios had significant mean difference with 1PC mechanism producing lower execution times, eight scenarios had either large or typical effect size, and all 12 scenarios had high or adequate power. Thus, we conclude that 1PC mechanism performed better than 2PC mechanism. We, further argue that 1PC

protocol optimization for WS-AT offers significant reduction in the execution time for transactions involving a single participant in comparison to 2PC.

## Chapter 7

### CONCLUDING REMARKS

The current WS-Atomic Transaction specification supports only 2PC. This research has provided a support for 1PC mechanism in WS-AT and it is available for implementation in a single participant scenario. 1PC will optimize the web service transaction process in terms of overhead and improvement in performance in terms of execution time (a.k.a., response time). In a single participant scenario, the decision to either commit or rollback is solely based on the participant. By avoiding the “prepare” phase, the number of messages to exchange between a participant and Transaction Coordinator would be less when compared to that for the 2PC commit protocol. Based on experimental results, it can be concluded that there is a definite improvement in performance for the 1PC mechanism when compared to the 2PC mechanism in terms of execution time. Shorter execution time of 1PC mechanism can be helpful in providing fast and efficient online business transactions in one participant scenarios. Findings of this thesis have considerable implications for businesses as fast and efficient online transactions are known to maintain customer loyalty and affect the profitability of a business (Srinivasan, et al., 2002).

We have identified areas which can be considered for future research work. Enabling One-phase commit (1PC) will also prevent the failure due to the failed state of a Transaction Coordinator. The reason behind is that, tight coupling between Transaction coordinator and Participant is removed. This will remove the possibility of a failure of coordinator where

Participant cannot progress until the Transaction Coordinator recovers from the failure. The experiments used for evaluation in this thesis did not test for Transaction Coordinator failure contexts. We suggest as a future work to evaluate performance of 1PC mechanism for WS-AT for Transaction Coordinator failure scenarios. Further, based on the results of this thesis, it is recommended that OASIS should consider including 1PC mechanism in WS-AT specification and JBoss should consider incorporating 1PC mechanism in their JBoss Transactions technical implementation framework.

## REFERENCES

- Al-Houmaily, Y. J., & Chrysanthis, P. K. (2004). *1-2PC: The One-Two Phase Atomic Commit Protocol*. Paper presented at the ACM symposium on Applied computing, Nicosia, Cyprus.
- Benbasat, I., & Zmud, R. W. (1999). Empirical Research in Information Systems: the Question of Relevance. *MIS Quarterly*, 23(1), 3 - 16.
- Benbasat, I., & Zmud, R. W. (2003). The Identity Crisis Within the IS Discipline: Defining and Communicating the Discipline's Core Properties. *MIS Quarterly*, 27(2), 183-194.
- Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., & Orchard, D. (2004, 11 February 2004). Web Services Architecture - W3C Working Group Note Retrieved March 25, 2014, from <http://www.w3.org/TR/ws-arch/>
- Bowles, J., & Moschoyiannis, S. (2008). When Things Go Wrong: Interrupting Conversations. In J. L. Fiadeiro & P. Inverardi (Eds.), *Fundamental Approaches to Software Engineering* (Vol. 4961, pp. 131-145). Berlin Heidelberg: Springer
- BPMN. (2011, January 2011). Business Process Modeling Notation Specification Retrieved March 30, 2014, from <http://www.omg.org/spec/BPMN/2.0/>
- Ceponkus, A., Dalal, S., Fletcher, T., Furniss, P., Green, A., & Pope, B. (2002, 3 June). Business Transaction Protocol Retrieved March 25, 2014, from [https://www.oasis-open.org/committees/download.php/1184/2002-06-03.BTP\\_cttee\\_spec\\_1.0.pdf](https://www.oasis-open.org/committees/download.php/1184/2002-06-03.BTP_cttee_spec_1.0.pdf)
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd. ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Congiu, G., Grawinkel, M., Narasimhamurthy, S., & Brinkmann, A. (2012, 24-28 Sept. 2012). *One Phase Commit: A Low Overhead Atomic Commitment Protocol for Scalable Metadata Services*. Paper presented at the IEEE International Conference on Cluster Computing Workshops (CLUSTER WORKSHOPS), Washington, DC, USA.

- Constantinides, E. (2004). Influencing the online consumer's behavior: the Web experience. *Internet Research*, 14(2), 111-126.
- Dinn, A., Connor, K., & Little, M. (2014). Transactions XTS Administration And Development Guide Retrieved March 30, 2014, from [http://docs.jboss.org/jbosstm/5.0.0.M1/guides/xts-administration\\_and\\_development\\_guide/ch03.html](http://docs.jboss.org/jbosstm/5.0.0.M1/guides/xts-administration_and_development_guide/ch03.html)
- Faul, F., Erdfelder, E., Lang, A.-G., & Buchner, A. (2007). G\*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences *Behavior Research Methods*, 39(2), 175-191.
- Hevner, A., & Chatterjee, S. (2010). Design Science Research in Information Systems. In A. Hevner & S. Chatterjee (Eds.), *Design Research in Information Systems* (Vol. 22, pp. 9-22). New York, USA: Springer.
- Hirschheim, R., & Klein, H. K. (2003). Crisis in the IS Field? A Critical Reflection on the State of the Discipline. *Journal of the Association for Information Systems*, 4(1), 237-293.
- JBossTS. (2014). JBoss Transactions - Narayana Retrieved March 25, 2014, from <http://www.jboss.org/narayana>
- Kavantzaz, N., Burdett, D., Ritzinger, G., Fletcher, T., Lafon, Y., & Barreto, C. (2005, 9 November ). Web Services Choreography Description Language (WS-CDL) Version 1.0 Candidate Recommendation. Retrieved March 25, 2014, from <http://www.w3.org/TR/ws-cdl-10/>
- Little, M. (2007). JBoss Community - Transactions Discussions Retrieved March 25, 2014, from [https://community.jboss.org/message/513698?\\_sscc=t](https://community.jboss.org/message/513698?_sscc=t)
- Little, M., Maron, J., & Pavlik, G. (2004). *Java Transaction Processing: Design and Implementation* (1st ed.). Upper Saddle River, NJ, USA: Prentice Hall.
- Morgan, G. A., Leech, N. L., Gloeckner, G. W., & Barrett, K. C. (2007). *SPSS for Introductory Statistics: Use and Interpretation* (3rd ed.). New York, USA: Lawrence Erlbaum Associates.



- Neto, I. S., & Reverbel, F. (2008, 14-16 May). *Lessons Learned from Implementing WS-Coordination and WS-AtomicTransaction*. Paper presented at the IEEE/ACIS International Conference on Computer and Information Science, Washington, DC, USA.
- Newcomer, E., Robinson, I., Feingold, M., & Jeyaraman, R. (2009, 2 February 2009). Web Services Coordination (WS-Coordination) Version 1.2 Retrieved March 25, 2014, from <http://docs.oasis-open.org/ws-tx/wstx-wscoor-1.2-spec-os/wstx-wscoor-1.2-spec-os.html>
- Newcomer, E., Robinson, I., Freund, T., & Little, M. (2007, 12 July). Web Services Business Activity (WS-BusinessActivity) Version 1.1 Retrieved March 25, 2014, from <http://docs.oasis-open.org/ws-tx/wstx-wsba-1.1-spec/wstx-wsba-1.1-spec.html>
- Newcomer, E., Robinson, I., Little, M., & Wilkinson, A. (2009, 2 February). Web Services Atomic Transaction (WS-AtomicTransaction) Version 1.2 Retrieved March 25, 2014, from <http://docs.oasis-open.org/ws-tx/wstx-wsat-1.2-spec-os/wstx-wsat-1.2-spec-os.html>
- Onwuegbuzie, A. J., & Leech, N. L. (2004). Post Hoc Power: A Concept Whose Time Has Come. *Understanding Statistics*, 3(4), 201-230. doi: 10.1207/s15328031us0304\_1
- Orlikowski, W. J., & Iacono, C. S. (2001). Research Commentary: Desperately Seeking the "IT" in IT Research—A Call to Theorizing the IT Artifact. *Information Systems Research (ISR)*, 12(2), 121-134.
- Papazoglou, M. P. (2003). Web Services and Business Transactions. *World Wide Web*, 6(1), 49-91.
- Purao, S., Bagby, J., & Umapathy, K. (2008). *Standardizing Web Services: Overcoming 'Design by Committee'*. Paper presented at the IEEE Congress on Services - Part I, Hawaii, US.
- Shneiderman, B. (1984). Response time and display rate in human performance with computers. *ACM Computing Surveys*, 16(3), 265-285. doi: 10.1145/2514.2517
- Singhal, M. (1988). Issues and approaches to design of real-time database systems. *ACM SIGMOD Record*, 17(1), 19-33. doi: 10.1145/44203.44205

Spatz, C. (2010). *Basic Statistics: Tales of Distributions* (10th ed.). Belmont, CA, USA: Cengage Learning.

SPSS. (2014). IBM SPSS Statistics. Armonk, NY, USA: IBM. Retrieved from <http://www-01.ibm.com/software/analytics/spss/products/statistics/>

Srinivasan, S. S., Anderson, R., & Ponnnavolu, K. (2002). Customer loyalty in e-commerce: an exploration of its antecedents and consequences. *Journal of Retailing*, 78(1), 41-50. doi: [http://dx.doi.org/10.1016/S0022-4359\(01\)00065-3](http://dx.doi.org/10.1016/S0022-4359(01)00065-3)

Umapathy, K., Purao, S., & Bagby, J. (2012). Empirical analysis of anticipatory standardization processes: a case study. *Information Systems and E-Business Management*, 10(3), 325-350. doi: 10.1007/s10257-011-0169-1

Weber, R. (1987). Toward a Theory of Artifacts: A Paradigmatic Base for Information Systems Research. *Journal of Information Systems*, 1(2), 3-19.

WS-CAF. (2005). Web Services Composite Application Framework (WS-CAF) Retrieved March 25, 2014, from [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=ws-caf](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-caf)

## VITA

Chirag N. Rana is currently working as a Lead Software Developer in the IT Department of one of the leading health insurance firms in Jacksonville, Florida. He has more than ten years of experience in the IT industry mainly in Java/J2EE Applications and Portal Applications (Enterprise Portal and Web Content Management). He also has hands-on exposure to Service Oriented Architecture (SOA) and web services development. He has specialized in IBM WebSphere Portal and IBM Web Content Management System. He holds a professional IBM WebSphere Portal Solution Developer certification. He holds IT degrees viz. Master of Business Administration in IT (MBA-IT), Master's Diploma in Computer Applications, and Advanced Diploma in Computer Applications. He is a self-motivated, team player as well as an independent worker. He is always ready to learn and adapt new tools and technologies. He is originally from India and currently lives in Jacksonville, Florida.